3 Top Verification Challenges for the Next 5 Years

Verification Futures – Sophia Antipolis

Laurent Arditi

ARM, Processor Division

Sophia-Antipolis, France

November 2013

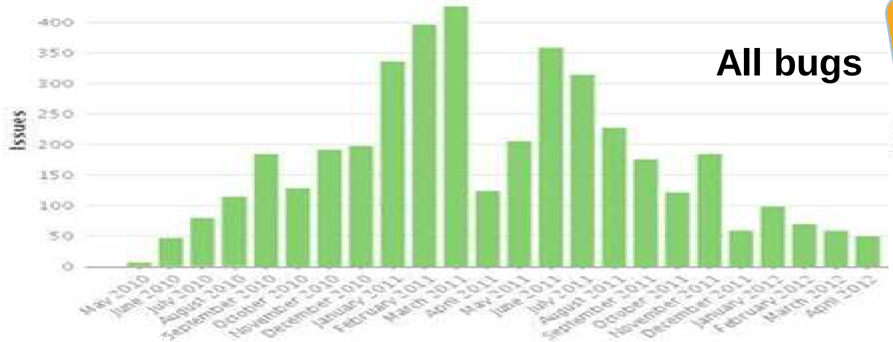
Functional Verification Today at ARM

- Mostly simulation driven in the first place
  - Various block-level testbenches (constrained-random)
  - Top-level validation suites
  - RTL vs. model comparisons
- Emulation, FPGA and silicon
  - OS boot and applications
  - Soak testing

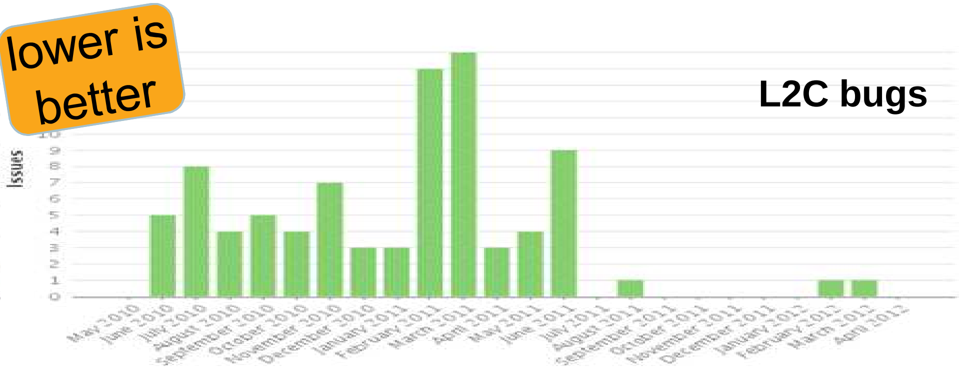
- Formal verification usage increasing very fast
- The ARM AHAA terminology:
  - Bug Avoidance: designing correct things
  - Bug Hunting: finding as many bugs as possible
  - Bug Absence: ensuring there's no bug
  - Bug Analysis: characterize known bugs and validate fixes

Challenge #1: Bug Avoidance

- How to apply more bug avoidance strategies to the development of complex IPs?
- How to design IPs which are fundamentally correct?
- Formal design bring-up shows good results:
  - Formal methods used directly by the design team, at the block-level, and very early (before the availability of the simulation testbench)
  - The bugs are found much earlier, the checked-in design is cleaner, and simulation can focus on higher quality bugs
  - Example GPU where L2C used formal bring-up:



All bugs



lower is better

L2C bugs

Challenge #2: Bug Hunting

- Scalability of simulation
  - Can we grow our simulation farms forever?
- How to make a better usage of emulation/FPGA/silicon?
  - The ability of debug is key here

- Formal is a good complement, but how does it scale?
  - Is brute-force the answer?
  - Is a formal soak efficient?
  - How to take benefits from simulation results?

- Measurement and sign-off are keys
  - When can we say “done”?
  - Which coverage metrics are meaningful?
  - How to correlate and merge coverages from simulation and formal?

Challenge #3: Bug Absence

- Bug absence techniques has a much higher value compared to bug hunting
- But they are also very costly (talking about human cost)
- How to define completeness?
- Complexity scalability
  - Many configurations
  - Infinite end user cases
- Proliferation of complex products into all aspects of everyday life
  - ➢ Considerations for safety and reliability
- Deployment of formal technologies is key
  - In theory the only technique to achieve bug absence